UNITED STATES PATENT APPLICATION

FOR A

METHOD AND SYSTEM FOR CACHING SECURE WEB CONTENT

Inventors:

Rajeev Chawla, Panagiotis Tsirigotis and Dan Boneh

ATTORNEY DOCKET NO. 24631.710

Please direct communications to :

WILSON SONSINI GOODRICH & ROSATI
650 Page Mill Road
Palo Alto, CA 94304
(650) 493-9300

Express Mail Number EL 757543096 US

METHOD AND SYSTEM FOR CACHING SECURE WEB CONTENT

RELATED APPLICATIONS

This application claims the benefit of United States Provisional Application Number 60/223,171 filed on 08/07/2000, which is incorporated herein by reference.

FIELD OF THE INVENTION

The field of the invention is secure content in a network system. More particularly the invention is in the field of secure content transfer in a network using caching techniques.

BACKGROUND OF THE INVENTION

Web caches are network applications used to reduce network traffic and improve response times. Web caches work by storing static content on a network at intermediate locations. Static content encompasses items that rarely change. Once stored the information is available for repeated transmissions of the same content over an abbreviated portion of the network. By eliminating the need for the server to produce all of the requested information for each request, the effective bandwidth of the network is increased. Unfortunately, web caching does not currently apply to secure web content. Secure web content is sent encrypted using various protocols such as Transport Layer Security ("TLS") or Secure Socket Layer ("SSL"). Such secure protocols use unique encryption keys known only to the connection endpoints. Each message therefore is independently secure. Consequently, intermediate web caches in a computer network do not store and retransmit secure

static content since they cannot examine it to determine if it has changed. As far as the cache is concerned each message is unique effectively eliminating the purpose of a cache entirely. Hence, TLS and SSL are incompatible with the existing web caching architecture.

Transport Layer Security protocol is one of the most widely deployed protocols for securing communications on the World Wide Web ("WWW") and is used by most E-commerce and financial web sites. It guarantees privacy and authenticity of information exchanged between a web server and a web browser. Currently, the number of web sites using TLS or SSL to secure web traffic is growing at a phenomenal rate. As the services provided by the World Wide Web continue to expand, so will the need for security. Unfortunately, TLS and SSL are incompatible with the current network design methodologies used in the Internet.

This incompatibility stems from the inherent nature of how a secure session is established. A TLS session, for example, between a web server and a web browser occurs in a number of phases. When a web browser first connects to a web server using TLS, the browser and server execute the TLS handshake protocol. The outcome of this protocol is a session encryption key and a session integrity key. These keys are known only to the web server and the web browser.

Once the session keys are established, the browser and server begin exchanging data. The data is encrypted using the session encryption key and protected from tampering using the session integrity key. When the browser and server are done exchanging data the connection between them is closed. If the browser and server subsequently reestablish a secure connection the browser and server may execute a resume handshake or establish a new set of session keys. A

resume handshake protocol causes both server and browser to reuse the session key previously established during the initial handshake, and is more efficient, but requires the connection between the web server and web browser to be continuous. Thereafter, all application data is encrypted and protected using the previously established session keys.

Web caches are typically located on the network between the user and the web server being accessed. The web cache inspects all responses coming back from the server and stores in its memory all content that changes infrequently. This information is called static content. Examples of static content include the banner and the navigation buttons on the page. The next time a user requests this information the cache responds immediately with the information without contacting the web server. As a result, web caches dramatically reduce traffic on the network and reduce the response times to user requests.

A reverse proxy is similar to a web cache. The difference lies in where the reverse proxy is located and the type of content cached. While web caches are located close to the client processor so as to minimize response time, the reverse proxy is typically located close to the web server with the most common location being at the same site as the web server. The main goal of the reverse proxy is to reduce the load on the web server. Any time a request is received at the web site the reverse proxy first determines whether the response is already cached. If so, the reverse proxy responds itself without contacting the web server. Otherwise the request is sent to the web server. Inherent to the reverse proxy function is its ability to examine the request as well as the content of the cache to determine if the information stored fulfills the request.

Web caches and reverse proxies are ineffective when dealing with secure content. The problem lies in identification of repeated information. Secure content passing through these appliances is encrypted using a key known only to the end points, namely the web server and the web browser. Each web browser connected to the proxy passes through information that is unrecognizable to the cache. The web cache or the reverse proxy cannot interpret the data to determine if the data should be stored or if the data request matches any stored data. Hence it is useless to cache the encrypted information. Consequently, the existing infrastructure designed to make the Internet more efficient and faster becomes ineffective when dealing with secure content.

SUMMARY OF THE INVENTION

A method and system are provided for caching secure content on a computer network. One embodiment establishes a reverse proxy logically located between a web server and connections to the outside world that is capable of interpreting and storing secure content.

The Secure Reverse Proxy ("SRP") in one embodiment intercepts request for secure content prior to the demand being received by the web server. The SRP establishes an encrypted session with the web browser to facilitate the SRP's ability to examine the secure content. Once the secure request is decrypted, the SRP examines its cache and determines if the requested content available. If the requested content is available, the SRP encrypts it using the established session keys with the web browser and transmits the information. In this embodiment the web

browser never directly contacts the web server nor does the web server need to respond to the request.

In an additional embodiment the SRP determines if the requested information is not available in the SRP's cache. Upon determining that the information is not cached, the SRP establishes a secure connection with the web server using TLS, SSL or other secure protocol. The SRP forwards the web browser's request for information to the web server and the web server responds to the SRP as if it was the web browser. Upon receiving the information the SRP stores it in the cache for future use in either encrypted or clear-text form. With the information requested by the browser now available in the SRP's cache, the SRP retrieves the information and encrypts it using the session keys established between the SRP and the web browser. The SRP then transmits the information to the web browser. Since the requested information is now contained in the SRP's cache, subsequent requests from this or other browsers for the same information will not require any interaction with the web server. Thus the efficiency of the network is increased and the load upon the server is diminished.

BRIEF DESCRIPTION OF THE FIGURES

The present invention is illustrated by way of example in the following diagrams and flow charts in which like references indicate similar elements. The following diagrams and flow charts disclose various embodiments of the present invention for purposes of illustration only and are not intended to limit the scope of the invention.

Figure 1 is a block diagram of an embodiment of a network system for improving secure communications.

Figure 2 is a flow diagram illustrating a method for secure reverse proxy caching of secure content for one embodiment of the present invention.

DETAILED DESCRIPTION

A method and system are provided for secure reverse proxies capable of caching secure content. These Secure Reverse Proxies ("SRP") are, in one embodiment, installed logically between the web server and the connection to the outside world with all incoming secure requests being first sent to the SRP rather than directly to the server.

Normally, a request to establish a secure connection is received by the server from a web browser. In one embodiment the request message, such as a request to establish a TLS session, is referred to as a client hello and is directed to the SRP instead of the server. The SRP responds to the request by sending back a TLS server hello message containing the server certificate. After performing the TLS key exchange protocol the SRP and the requesting browser share a secret encryption key and a secret integrity key. These keys are used to protect the rest of the session and are appropriately called session keys.

With a TLS session established between the SRP and the web browser the requesting web browser sends an encrypted HTTP request using the TLS protocol and the session keys. The SRP receives the request, decrypts it using the session keys and checks whether the local cache contains an appropriate response to the incoming inquiry. If the SRP contains the requested information, the SRP encrypts the response using the current session keys and sends the result to the requesting web browser. Up to this point the web server has never seen the request.

If the request received by the SRP is not contained in the local cache the SRP must forward the request to the web server. This requires the SRP and the web server to establish a secure session, independent of the secure session between the

SRP and the web browser. The request to the web server is thereafter secured using a secure protocol such as TLS so that is does not appear as clear-text on the network. In another embodiment all communication between the SRP and the web server takes place over a private network at the web site or by an inherently secure connection such as fiber optic or copper cable. If such a connection between the server and the SRP is present it is possible for the SRP to send the request to the web server using clear-text HTTP. This reduces the load on the web server since the web server need not perform any expensive secure handshake computations.

With communication between the web server and the SRP established either through a secure network protocol or an inherently secure connection, the web server sends a response to the query back to the SRP. Having received a response from the web server, the SRP caches the information in either cipher or clear-text format. Now possessing the requested information in the local cache, the SRP encrypts the requested information using the current SRP / browser session keys and sends the result to the requesting web browser. While the SRP can maintain numerous secure connections with a number of web browsers using a number of session keys, the SRP need only maintain one secure connection with the server. Thus the load on the server is dramatically reduced and the efficiency of the network is significantly improved.

In one embodiment the SRP stores the cached information locally within the server or on a non-volatile medium such as magnetic tape, optical disks, by a third party, or using other techniques known in the art. To ensure the information remains secure, information is stored on non-volatile mediums encrypted under a separate

key known only to the server. The server maintains the key to the information using a tamper resistant non-volatile card.

Figure 1 shows one embodiment for enhancing secure network communications. The system 100 includes multiple client computers 132, 134, 136, and 138, which are coupled to a server system 110 through a network 130. The network 130 can be any network, such as a local area network, a wide area network, or the Internet. Coupled among the server system 110 and the network 130 is a Secure Reverse Proxy 150. While shown as a separate entity, the SRP 150 can be located independently of the server system, the network environment or distributed among any number of server sites 112, 114 and 116. The client computers each include one or more processors and one or more storage devices. Each of the client computers also includes a display device, and one or more input devices. The SRP can be one or more devices, each including one or more processors and storage devices.

All of the storage devices store various data and software programs. In one embodiment, the method for improving TLS is carried out on the system 100 by software instructions executing on one or more of the server sites 112, 114 and 116. The software instructions may be stored on the server system 110 any one of the server sites 112 - 116 on any one of the client computers 132 - 138 or any number of SRPs. For example, one embodiment presents a hosted application where an enterprise requires secure communications with the server. The software instructions to enable the communication to be cached by the SRP are stored on the server. In other embodiments, the software instructions and the caching process may be stored and executed on the client computers. Data required for the execution

of the software instructions can also be accessed via the network and can be stored anywhere on the network.

Figure 2 is a flow diagram for enhancing secure content on a network using reverse proxies of an embodiment. The process begins with a request by the web browser to establish a secure connection 210. The SRP responds with a hello message 220 and a TLS key is exchanged and validated between the browser and the SRP 230. With the establishment of the secure session 240 the SRP receives an HTTP request 250. The SRP examines the local cache 260 and if the content is not cached, forwards the request to the web server 270. This forwarding is through an independent TLS session established between the SRP and the web server. The web server responds with the secure content 280 which is locally cached at the SRP 285 and then forwarded to the web browser 290 via the earlier established session keys.

While TLS and other secure network protocols typically prevent the intermediate storing of secure static content on a reverse proxy, the architecture described herein enables such content to be cached. Hence, the web browser continues to receive encrypted content yet requests for encrypted material at the server are significantly reduced. The secure protocol between the SRP and the web browser preserves the confidentiality of the communication, as does the connection between the SRP and the web server. This connection provides no clear-text traffic from which an eavesdropper or active attacker can gain information. In addition to this security benefit there is also a significant performance advantage since the web server responds to minimal browser requests.

Performance is also enhanced by reducing the costly protocol of establishing a secure connection. When an SRP is not used, the web server must perform an

initial TLS or equivalent handshake with every new user that connects to the site. The TLS handshake is expensive and slows down the network. In one embodiment, the web server sees only connections from the SRP as the need to establish multiple TLS connections is displaced to the SRP. Having once established a secure connection, the SRP and the server can utilize a more efficient resume handshake rather than having to reinitiate the creation of a unique session key. Consequently, the need to perform and handle TLS handshakes repetitively is eliminated and only one TLS handshake with the SRP need be accomplished. The resulting reduced load on the sever increases effective network bandwidth and reduces cost.

From the above description and drawings, it will be understood by those of ordinary skill in the art that the particular embodiments shown and described are for purposes of illustration only and are not intended to limit the scope of the invention. Those of ordinary skill in the art will recognize that the invention may be embodied in other specific forms without departing from its spirit or essential characteristics.